HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

OIPE JUL 22 2005 PATENT & TRADEMARK OFFICE

PATENT APPLICATION

ATTORNEY DOCKET NO. 10016639-1

AF$ IFW

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Donald C. Soltis, Jr., et al. **Confirmation No.:** 6701

Application No.: 10/074,061 **Examiner:** Daniel H. Pan

Filing Date: Feb. 11, 2002 **Group Art Unit:** 2183

Title: STACKED REGISTER ALIASING IN DATA HAZARD DETECTION TO REDUCE CIRCUIT

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on May 20, 2005.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$500.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: July 20, 2005
OR
( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number______ on ______

Number of pages:

Typed Name: Janet Ridpath

Signature: Janet Ridpath

Respectfully submitted,

Donald C. Soltis, Jr., et al.

By Curtis A. Vock

Curtis A. Vock
Attorney/Agent for Applicant(s)
Reg. No. **38,356**

Date: **July 20, 2005**

Telephone No.: **(720) 931-3011**

PATENT
Attorney Docket No.: 10016639-1

IN THE UNITED STATES PATENT OFFICE

Applicant(s)	Donald C. Soltis Jr.	Examiner	PAN, DANIEL H
Serial No.	10/074,061	Group Art No.	2183
Filed	February 11, 2002	Confirmation No.	6701
For	Stacked Register Aliasing in Data Hazard Detection to Reduce Circuit		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

In accord with 37 CRF § 41.37, and fully responsive to the Office Action of February 22, 2005, Appellants hereby file their appeal brief in support of their Appeal in the above-identified matter (hereinafter the '061 Application). A notice of appeal, with appropriate fee of $500 as required by §§41.31, 41.20(b)(1), was filed on May 20, 2005. The $500 fee for this appeal brief, as required by 37 CRF §41.20(b)(2), is also filed herewith. This appeal brief is timely filed within two months of the mailing of the notice of appeal.

(1) **Real party in interest.**

The real party in interest for this appeal is Hewlett-Packard Development Company, L.P. (HPDC), a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249, Houston, TX 77070, U.S.A. HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, L.L.C. Evidence of this assignment, which was recorded on June 18, 2003, may be found at reel/frame 013776/0928.

(2) Related appeals and interferences.

No other appeals or interferences are currently known to Appellants that will directly affect, be directly affected by, or have a bearing on the decision to be rendered by the Board of Patent Appeals and Interferences in the present appeal.

(3) Status of claims.

Claims 1-16 are pending in the '061 Application. Applicants appeal all claims 1-16.

Claims 1, 3, 8-12, 15 and 16 stand rejected under 35 U.S.C. §102(a) as being anticipated by U.S. Patent No. 5,884,070 ("Panwar").

Claims 2, 13 and 14 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Panwar in view of U.S. Patent No. 5,706,478 ("Dye").

Claims 4 and 5 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,826,055 ("Wang") in view of U.S. Patent No. 6,598,149 ("Clift").

Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,371,684 ("Iadonato") in view of Clift.

(4) Status of amendments.

The '061 Application was filed on February 11, 2002. A first office action was mailed on September 7, 2004, to which a response was filed and entered December 7, 2004. On February 22, 2005, a final office action was mailed, and on April 8, 2005, an after final amendment was filed in response thereto. On April 22, 2005, an advisory action was mailed, prompting a notice of appeal, filed on May 20, 2005. Claims 1-16 are currently pending, of which claim 7 is original (without claim amendment during prosecution). Claims 1-6 were amended and claims 8-16 were added in the response of December 7, 2004.

(5) Summary of claimed subject matter.

The '061 Application discloses, for example, register aliasing for data hazard detection logic of a processor that "simplifies the logic associated with ... data hazards so that a virtual register file may map frames of data to a physical register file of equal or larger size ... without corresponding growth of data hazard

detect logic." See paragraph 11 of the '061 Application. Complexity within the data hazard detection logic is reduced by aliasing multiple physical register IDs to one register ID within the data hazard detection logic. See paragraph 22 and FIG. 4. For example, FIG. 4 illustrates thirty-two register IDs, RID(32)-RID(63) within the data hazard detection logic that alias to one-hundred-and-twenty-eight general registers, GR(32)-GR(159), of the processor. Thus, in this example, each register ID within the data hazard detection logic is aliased to four general registers, such that the data hazard detection logic need only compare thirty-two register IDs and not one-hundred-and-twenty-eight. Thus, the complexity of the data hazard detection logic is reduced, since fewer comparisons are required.

Accordingly, claims 1-3 relate to a method for stacked register aliasing in a data hazard detection of a processor. Claims 4-6 relate to a processor for processing program instructions. Claim 7 relates to data hazard detection logic of a processor of the type having a register file and a register ID file providing row-to-row data hazard detection. Claims 8-12 relate to a method for data hazard detection within a processor. Claims 13 and 14 relate to a method of reducing data hazard logic dependency on size of a register file within a processor. Claim 15 relates to a method of data hazard detection within a processor. Claim 16 relates to a method for stacked register aliasing in data hazard detection logic of a processor.

(6) <u>Grounds for rejection to be reviewed on appeal</u>.

A. Whether claims 1, 3, 8-12, 15 and 16 are anticipated by Panwar in accordance 35 U.S.C. §102(a) and (b).

B. Whether claims 2, 13 and 14 are unpatentable over Panwar in view of Dye in accordance 35 U.S.C. §103(a).

C. Whether claim 7 is unpatentable over Iadonato in view of Clift in accordance 35 U.S.C. §103(a).

D. Whether claims 4 and 5 are unpatentable over Wang in view of Clift in accordance 35 U.S.C. § 103(a).

(7) **Arguments.**

Argument A

Panwar does not teach or suggest the elements of claims 1, 3, 8-12, 15 and 16. These claims do not stand or fall together. Generally, the '061 Application discloses register aliasing with data hazard detection logic of a processor. In an example of operation, two or more groups of registers of a stacked register file are aliased to one group of register IDs within the logic. Data hazards within registers of the multiple groups of registers are then detected by comparing register IDs; each register ID aliasing to one register within each group.

This is not what Panwar discloses. Instead, Panwar discloses that "for single precision operations using aliased registers, there are at least four possible dependencies per instruction since each source register can have two possible dependencies." See Panwar col. 4, lines 11-14. Panwar does not disclose reducing complexity of hazard logic through use of aliasing. Rather, Panwar discloses that "the single precision instruction, which could normally have up to four dependencies, is divided into two separate microinstructions wherein each microinstruction would have possibly only two data dependencies." See Panwar col. 6, lines 4-8. Panwar thus concerns aliasing of two single precision registers to one double precision register, and only provides for the specific case of single precision instructions. Panwar only reduces four dependencies to two dependencies for single precision instructions.

Panwar discloses that each consecutive pair of single length registers forms a 'double-length' register and the register IDs of these single length registers are aliased. Specifically, "when two or more data addresses refer to the same datum, the address is said to be 'aliased'." See Panwar col. 2, lines 45-47. However, in the '061 Application, aliased addresses do not refer to the same datum, and consecutive registers are NOT aliased! As shown in the example of paragraph [0022], general registers 32, 64 and 96 are aliased, general registers 33, 65 and 97 are aliased, and so on. The aliasing of Panwar is different to the aliasing of the '061 Application.

More specifically, the aliasing of Panwar refers to a mechanism of identifying consecutive register addresses (e.g., f4 and f5) that may reference a common datum.

In the example of Panwar, single length register f5 addresses the same datum as double register f4, which addresses locations f4 and f5, and, therefore, in Panwar, f4 and f5 are aliased. See Panwar background, and FIGs. 1A and 1B. Panwar thus utilizes aliasing for register identifiers that address the same datum.

There is no disclosure in Panwar of groups of registers utilizing common register IDs and overlapping in data hazard detect logic (as for example described in independent claims 1, 8, 13, 15, 16). According to the '061 Application, aliasing within hazard detection logic enables, for example, 32 register IDs to map a large register file into 32 register sequences (claim 6) and yet with the same normal 32-row data hazard logic as before.

The Examiner refers to FIG. 1A and 1B of Panwar; but these figures clearly do not disclose the use of register IDs as claimed. Panwar's FIG. 1A and 1B are discussed in the 'background art' and specifically illustrate how two single precision registers (e.g., f5 and f4) alias to a single double precision register f4 (see also Panwar, column 4, lines 9-15). According to Panwar, aliasing of two or more addresses refer to the same datum. The register pairs of Panwar cannot be considered as groups of registers as required by the '061 Application, at least for the reason that Panwar aliases these register pairs together, and not across groups as required by the '061 Application.

And Panwar also does not disclose – anywhere - how data hazard detect logic is overlapping in the detection of hazards through the register IDs. To further illustrate this point, Panwar could instead utilize the background (prior art) hazard detect logic and schema shown in FIG. 1 and FIG. 2 of the '061 Application.

Accordingly, there is no disclosure within Panwar of the following features:

- Claim 1 - "the first and second register identifiers overlap in hazard detect logic across two or more rows of the register file."
- Claim 8 - "aliasing each register identifier of a group of register identifiers to two or more registers of a register file of the processor."
- Claim 13 - "evaluating matches between entries of the register ID file in the hazard logic without distinguishing between common aliased entries of the register file."

- Claim 15 - "aliasing each register ID within data hazard detection logic to two or more registers of a register file."
- Claim 16 - "aliasing two or more groups of registers of a stacked register file to one group of register IDs within the data hazard detection logic."

In particular, claim 1 requires that "the first and second register identifiers overlap in hazard detect logic across two or more rows of the register file." Thus the first and second register identifiers overlap in hazard detect logic, but actually address two or more rows of the register file, whereas Panwar discloses that the register identifier alias the same datum. Therefore, Panwar cannot anticipate claim 1.

With regard to claim 2, therefore, Panwar also fails to disclose utilizing groups of 32 register identifiers to alias data hazard detect logic to windows of 32-register frames.

Claim 8 requires "aliasing each register identifier of a group of register identifiers to two or more registers of a register file of the processor." Each register identifier is thus aliased to two or more datum (i.e., actual registers) and not one as required by Panwar. Therefore, Panwar cannot anticipate claim 8.

In the context of claim 8, therefore, Panwar also fails to disclose features of claims 9-12, including:

- the group of register identifiers maps to two or more non-overlapping groups of registers of the register file (claim 9).
- the step of determining utilizes data hazard detection logic corresponding to two or more rows of the register file (claim 10).
- the step of aliasing increases the register file size without a corresponding increase in data hazard detection logic (claim 11).
- the group of register identifiers has 32 register identifiers (claim 12).

Claim 13 requires "evaluating matches between entries of the register ID file in the hazard logic without distinguishing between common aliased entries of the register file." Thus, the multiple entries addressed by the register ID are not distinguished when evaluating the register ID within the hazard detection logic. Since

Panwar does not identify multiple data locations with a register ID, Panwar cannot anticipate claim 13.

Claim 15 requires that each register ID is aliased within data hazard detection logic to two or more registers of a register file. Panwar does not disclose or suggest the possibility of aliasing additional registers as required by claim 15. Therefore, Panwar cannot anticipate claim 15.

Claim 16 requires that two or more groups of registers of a stacked register file are aliased to one group of register IDs within the data hazard detection logic. Panwar on the other hand discloses aliasing of two consecutive registers, which cannot be interpreted as aliasing of two groups of registers. Panwar does not therefore suggest aliasing more than two register IDs. The aliasing of two groups of registers to one group of register IDs is a distinctly unique feature of the '061 Application. Therefore, Panwar also cannot anticipate claim 16.

Argument B

Dye was cited with Panwar, presumably, because Dye discloses a 128-register register file. The Examiner then contends that a combination of Panwar with Dye renders claims 2, 13, 14 obvious. However, this combination still fails to disclose the use of overlapping data hazard detect logic based on register IDs as in the present claims. Dye instead discloses a processor that "executes display list commands in processor and coprocessor mode and dynamically switches between these two modes." See Dye col. 3, lines 29-32. Dye does not disclose data hazard detection logic within the processor, nor does Dye teach or suggest aliasing of registers. Therefore, Dye and Panwar still cannot teach or suggest all claim elements of claims 1 and 13, as argued above. Panwar in view of Dye cannot then render claims 2, 13 and 14 obvious. Claims 2, 13 and 14 do not stand or fall together.

The Examiner contends that Panwar does disclose the data hazard detect logic of the present claims; however a thorough and complete review of Panwar will not reveal any such detect logic. Panwar's illustration that single precision registers may be "aliased" to a double precision register is not equivalent to the present claims.

According to claim 2, each of the steps of identifying includes identifying registers within a 128-register register file. Panwar and Dye do not disclosure this feature in the context of claim 1.

According to claim 13, a method for reducing data hazard detect logic includes selecting a register ID file size; aliasing at least one entry of the register ID file to two or more registers of the register file; and evaluating matches between entries of the register ID file in the hazard logic without distinguishing between common aliased entries of the register file. According to the foregoing arguments, Panwar and Dye do not disclosure features of claim 13.

According to claim 14, the step of aliasing increases the register file size without a corresponding increase in data hazard detection logic. Panwar and Dye do not disclosure this feature in the context of claim 13.

Argument C

Claim 7 recites that, in data hazard detect logic of a processor of the type having a register file and a register ID file providing row-to-row data hazard detection, an improvement is provided wherein the register file ID aliases row-to-row hazard detection of the register file by common data hazard detection logic for two or more rows of the register file. In an example of row to row hazard detection of claim 7, hazards are detected by identifying matching register IDs of registers within the register file. Claim 7 specifically teaches that the register file ID *aliases* row-to-row hazard detection of the register file for two or more rows of the register file. The register file ID aliases two or more registers of the register file. When the data hazard detection logic compares a first register ID to a second register ID, it effectively compares two or more rows of the register file (aliased by the first register ID) to two or more rows of the register file (aliased by the second register ID). That is to say, given a register identifier for one row of the register file, data hazard detection logic does not distinguish between any of the aliased rows within the register file when comparing row-to-row.

For example, again consider paragraph 22 of the '061 Application, which describes this example: "the register ID file has 32 register identifiers, then each subsequent set of 32 GRs beginning with GR(32) (e.g., GR(32:63), GR(64:95),

GR(96:127) and GR(128:159)) alias respectively to the same 32 hazard detect register identifiers RID(32:63), as illustrated in FIG. 4." Paragraph 22 continues with "register IDs now alias to common hazard detect logic for rows GR(32), GR(64), GR(96), for rows GR(33), GR(65), GR(97), and so on, of register file 114." Thus, in this example, it can be seen that register identifier RID(32) alias to registers GR(32), GR(64), GR(96) and GR(128), such that hazard detection logic does not distinguish between registers GR(32), GR(64), GR(96) and GR(128) when comparing row-to-row; hazard detection logic would identify a hazard when comparing register GR(32) to register GR(128), in this example.

In FIG. 2, Iadonato discloses a logic matrix that implements address comparison wherein "all source register addresses are compared with all previous destination register addresses." See Iadonato col. 5, lines 34-36. Thus Iadonato does not teach of reducing hazard detection logic complexity by comparing groups of *aliased* rows, as in claim 7. Clift too does not disclose this and, therefore, Iadonato in view of Clift cannot render claim 7 obvious.

Argument D

Wang and Clift do not disclose claims 4 and 5. These claims do not stand or fall together.

Claim 4 recites a processor for processing program instructions, including:

a) a register file;

b) an execution unit having an array of pipelines for processing the instructions and for writing bypass data to the register file; and

c) data hazard detect logic for detecting and aliasing data hazard detection for two or more rows of the register file.

In the '061 Application, rows of register files are aliased such that the data hazard detection logic applies to aliased register IDs. See paragraphs 11 and 22 of the '061 Application. Wang discloses that "data dependency logic, residing in RRC 204, is used for checking instructions for dependencies, and that "in checking for dependencies, the data dependency checking logic looks at the various registers to

determine whether one or more previous instructions must be executed before a subsequent instruction may be executed." See Wang col. 6, lines 58-63. Wang does not disclose or suggest, at least, aliasing as required by element c) of claim 4. Clift discloses a technique for enhancing performance for code transitions of floating point and packed data modes." See the abstract of Clift. Since neither Wang nor Clift teaches aliasing, they cannot be reasonably combined to render claim 4 obvious under 35 U.S.C. §103.

Claim 5 depends from claim 4 and benefits from like arguments; but in addition claim 5 has other features patentable over Wang and Clift. For example, claim 5 recites a register ID file for facilitating data hazard detection associated with rows of the register file, the register ID file having a plurality of register identifiers, the data hazard detect logic aliasing data hazard detection according to mapping of the register identifiers. Again, claim 5 requires that the register ID file facilitates data hazard detection. As shown in FIG. 4 and described in paragraphs 11 and 22 of the '061 Application, the register ID file does not have a one to one relationship with the register file of the processor. In fact, each entry within the register ID file is aliased to two or more registers of the register file. Since data hazard detection logic processes the register ID file, and the register aliases, complexity of the data hazard detection logic is reduced. Thus Wang and Clift again cannot render claim 5 obvious.

(8) Claims Appendix.

Appellants enclose a copy of the claims involved in this appeal as an appendix hereto.

(9) Evidence Appendix.

No evidence is entered or relied upon in this appeal.

(10) Related Proceedings Appendix.

To Appellants' knowledge, there are no decisions rendered by a court or the Board for submission with this appeal.

Conclusions

Appellants respectfully submit that the claims 1-16 patentably distinguish over the art of record. Other than the costs for the appeal brief, we believe no additional fees are due in connection with this matter. However, if any additional fee is deemed necessary, the Commissioner is hereby authorized to charge such fee to Deposit Account No. 08-2025.

Respectfully submitted,

By: Curtis A. Vock

Curtis A. Vock, Reg. No. 38,356
LATHROP & GAGE L.C.
4845 Pearl East Circle, Suite 300
Boulder, CO 80301
Telephone: (720) 931-3011
Facsimile: (720) 931-3001

CLAIM APPENDIX TO APPEAL BRIEF

1. (Previously Presented) A method for stacked register aliasing in data hazard detection of a processor, comprising the steps of:

identifying a first group of registers within a register file of the processor;

aliasing the first group of registers to first register identifiers;

detecting data hazards, if any, associated with the first register identifiers;

identifying a second group of registers within the register file;

aliasing the second group of registers to second register identifiers; and

detecting data hazards, if any, associated with the second register identifiers, wherein the first and second register identifiers overlap in hazard detect logic across two or more rows of the register file.

2. (Previously Presented) The method of claim 1, each of the steps of identifying comprising identifying registers within a 128-register register file.

3. (Previously Presented) The method of claim 2, the steps of detecting comprising utilizing groups of 32 register identifiers to alias data hazard detect logic to windows of 32-register frames.

4. (Previously Presented) A processor for processing program instructions, comprising:

a register file;

an execution unit having an array of pipelines for processing the instructions and for writing bypass data to the register file; and

data hazard detect logic for detecting and aliasing data hazard detection for two or more rows of the register file.

5. (Previously Presented) The processor of claim 4, further comprising a register ID file for facilitating data hazard detection associated with rows of the register file, the register ID file having a plurality of register identifiers, the data hazard detect logic aliasing data hazard detection according to mapping of the register identifiers.

6. (Previously Presented) The processor of claim 5, the register ID file mapping sequential 32-registers with the common hazard logic to more than 32 stacked registers of the register file to alias in 32-register sequences.

7. (Original) In data hazard detect logic of a processor of the type having a register file and a register ID file providing row-to-row data hazard detection, the improvement wherein the register file ID aliases row-to-row hazard detection of the register file by common data hazard detection logic for two or more rows of the register file.

8. (Previously Presented) A method for data hazard detection within a processor, comprising:

aliasing each register identifier of a group of register identifiers to two or more registers of a register file of the processor; and

determining data hazards within the register file by processing one or more of the register identifiers.

9. (Previously Presented) The method of claim 8, wherein the group of register identifiers maps to two or more non-overlapping groups of registers of the register file.

10. (Previously Presented) The method of claim 8, wherein the step of determining utilizes data hazard detection logic corresponding to two or more rows of the register file.

11. (Previously Presented) The method of claim 8, wherein the step of aliasing increases the register file size without a corresponding increase in data hazard detection logic.

12. (Previously Presented) The method of claim 8, wherein the group of register identifiers has 32 register identifiers.

13. (Previously Presented) A method of reducing data hazard logic dependency on size of a register file within a processor, comprising:

selecting a register ID file size;

aliasing at least one entry of the register ID file to two or more registers of the register file; and

evaluating matches between entries of the register ID file in the hazard logic without distinguishing between common aliased entries of the register file.

14. (Previously Presented) The method of claim 13, wherein the step of aliasing increases the register file size without a corresponding increase in data hazard detection logic.

15. (Previously Presented) A method of data hazard detection within a processor, comprising:

aliasing each register ID within data hazard detection logic to two or more registers of a register file; and

determining data hazards by matching register IDs within the data hazard logic.

16. (Previously Presented) A method for stacked register aliasing in data hazard detection logic of a processor, comprising:

aliasing two or more groups of registers of a stacked register file to one group of register IDs within the data hazard detection logic;

detecting data hazards, if any, associated with a first and second register of the two or more groups by comparing a first aliased register ID of the first register to a second aliased register ID of the second register within the data hazard detection logic;

wherein each register ID aliases to one register of each of the two or more groups of registers, the two or more groups of registers overlapping in hazard detect logic.

Evidence Appendix

(BLANK)

Related Proceedings Appendix

(BLANK)